IJTRET "Voice of Innovative Minds"
doi.org/10.54473/IJTRET.2024.8204
International Journal of Trendy Research in Engineering and Technology
Volume 8 Issue 2 April 2024
ISSN NO 2582-0958

FPGA Implementation of SPI Protocol

Nalina H D[1], Harshitha B V[2], Krupashree K P[2], Himashree J[2], Bhoomika M[2]

[1]*Assistant Professor,*[2]UG student, Dept. of Electronics and Communication,
GSSS Institute of Engineering & Technology for Women, Mysuru, Karnataka, India

Received 15 April 2024 Received in revised form 20 April 2024 Accepted 28 April 2024
Available Online 21 May 2024

ABSTRACT

Field Programmable Gate Array (FPGA) are versatile devices that can be programmed to implement various digital circuits, including communication protocols like Serial Peripheral Interface (SPI). The Serial peripheral interface protocol is a synchronous serial communication interface that allows for full duplex communication between a master device and a multiple slave MOSI, MISO, SCLK, and SS. MOSI and MISO are data lines, while SCLK is the clock signal generated by the master to synchronize transfers. SS is used by the master to select individual slave devices. By implementing the SPI protocol in FPGA, to gain flexibility and customization, as FPGAs can be reprogrammed for different applications without the need for dedicated hardware changes. The FPGA implementation of the Serial Peripheral Interface (SPI) protocol involves a structured methodology. With the help of pipeline buffer and DAM technique power dissipation is performed effectively. Finally, integrate the FPGA SPI module into the larger system, ensuring seamless communication with other components or microcontrollers. This systematic approach ensures a reliable and tailored FPGA based SPI implementation.

Keywords— FPGA, SPI IP, VHDL, Synchronous serial communication, Microcontrollers.

I. INTRODUCTION

Serial Peripheral Interface (SPI) is a widely used synchronous serial communication protocol for short- distance communication. The SPI bus is implemented by a module called a synchronous serial port on a mi- croprocessor control unit connected to peripheral de- vices. It is a simple, full-duplex, 4-wire protocol that uses a master-slave architecture. The master device initiates the communication and controls the clock signal, while the slave device receives data and trans- mits data. Fast communication can be built using the SPI protocol with the combination of VLSI technol- ogy and high-speed SPI design. VLSI facilitates the design of efficient interfaces that connect the FPGA to the SPI bus, ensuring reliable data transfer. The business Motorola invented SPI (Serial Peripheral In- terface), a protocol for serial synchronous communi- cation regarding the communication between the slave and master. which is also used to provide com- munication between microcontroller and many de- vices which are additional and like external Analog to the Digital Converters, Digital to Analog Converters, and EEPROMs. Modern communication protocols are outdated. Two distinct Protocols are available: A) SPI; B) Inter-I2C. Both protocols are well designed for the communications between the Integrated Cir- cuits for communication with On-Board Peripherals.

SPI is most used protocol for both intra-chip and in- ter-chip and is used at low or medium speed of data- stream transfer. The quality of the SPI Interface Pro- tocol with a Single Master and Single Slave configu- ration is introduced in this work. It involves 16-bit data transfer and all required features and used for modern applications such as ASIC or SOC (System on Chip). System Verilog is used to demonstrate the functional correctness and coverage code of the SPI design, and Verilog is also used to write the RTL for synthesis and then the Verification architecture is written using System Verilog.

II. RELATED WORKS

In their paper, Anant hula Srinivasa et al[1] presented a high performance SPI-IP with one Master One Slave configuration and 8-bit communication that includes all the necessary features required by modern ASIC/SoC applications. The Designed SPI is used for communication between different peripherals with that of a processor in a SoC application.

Muhammad Bakir and Niju Rajan [2] paper titled "VLSI implementation of SPI and I2C communication protocol" focuses on the simulation of SPI and I2C protocols. They reported that with a lot of peripheral devices from a different vendor there is always a choice to be made between the I2C and SPI protocols.

To decrease the delay, Satish and Waghmare [3] have suggested a system that uses a pipelined buffer between the Master and Slave. This will allow for synchronisation and increase the speed of data transfer as the delay in Transfer of data declines.

In their research, Sukhwinder Kaur et al. [4] examined and talked about how SPI (master-slave) configurations are implemented on the Lattice and Xilinx FPGA families. According to their research, as compared to non-gated architecture, gated clock architecture significantly reduces dynamic power dissipation and latency.

SPI is used, according to Mohd Noor and A. Saparon [5], to transport data between integrated components in motion controllers. Unlike earlier SPI designs that could only handle continuous data transfer, this one is made to function in three major data transfer modes: burst, interleave, and continuous. ModelSIM is used to mimic the SPI, and the ALTERA DE2-70 board is used to implement it on a Cyclone II FPGA chip.

Jiayi Qiang et al. [6] have presented the structure and operation of the SPI communication bus, examined its four operating modes and timing structure, and implemented the SPI bus communication function on an FPGA using a state machine technique. The Verilog hardware description language is used to write the SPI module circuit, and the Vividado simulator is used to model the waveform. Following the study of the simulation waveform, the state machine method's viability is confirmed.

III. METHODOLOGY

The proposed system of SPI protocol consists of the master and slave in whom we have introduced a pipelined buffer in the bus MOSI and MISO. The system takes the 1-bit address as an input. The data that was transferred from the master is kept in the buffer. As a result, there is no data loss during data transfer. The system's transfer rate or data transmission speed was lowered due to the system gates' delays. Factors like data rate, communication mode (full duplex/half-duplex), and number of connected slave devices are considered. Additionally, the chosen FPGA family and its available resources will influence the design approach. Functionalities of the SPI master or slave module, including data width, clock frequency, chip select handling, and any error detection/correction mechanisms. The designed code undergoes rigorous simulation using HDL simulators. This helps identify and rectify any functional errors before deploying the design on the actual FPGA. Simulation tools allow testing with various data patterns and scenarios to ensure proper behaviour. This methodology provides a structured approach for implementing reliable and efficient SPI interfaces using FPGA.

A. Block diagram

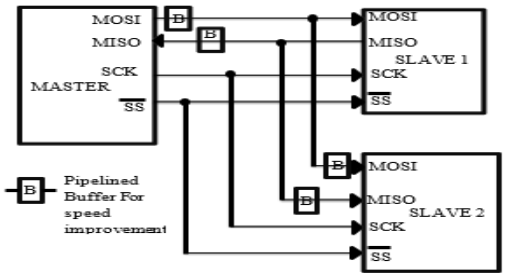

Fig 1.1: Block diagram of implementation of SPI protocol

B. Hardware and Software requirements

Hardware Requirements*:*

- FPGA (Field programable gate array) Software Requirements:
- Cadence tool

IV. RESULTS

Implementing the Serial Peripheral Interface (SPI) protocol using a pipeline buffer and Direct Memory Access (DMA) can lead to several benefits The pipeline buffer helps maintain a continuous flow of data between the SPI peripherals and memory, reducing the likelihood of bottlenecks and ensuring a steady data stream. DMA offloads the data transfer task from the CPU, allowing it to focus on other critical tasks. This can lead to more efficient use of system resources. Offloading data transfer tasks to DMA and utilizing a pipeline buffer can lead to faster response times, particularly in scenarios where real- time data processing is crucial to ease of implementation and reliability. response times, particularly in scenarios where real-time data processing is crucial. The outcomes of implementing the Serial Peripheral Interface (SPI) protocol depend on various factors, including the specific implementation details, hardware characteristics, and the requirements of the system. Here are some general expected

International Journal of Trendy Research in Engineering and Technology
Volume 8 Issue 2 April 2024
ISSN NO 2582-0958

outcomes: High-Speed Data Transfer: SPI is known for its high-speed serial communication, allowing for rapid data transfer between devices. Implementing SPI can lead to efficient and fast communication, especially in scenarios requiring quick exchange of data. Simple and Robust Communication: SPI has a straightforward protocol with master-slave architecture, enabling simple and robust communication between devices. This simplicity often translates to ease of implementation and reliability.

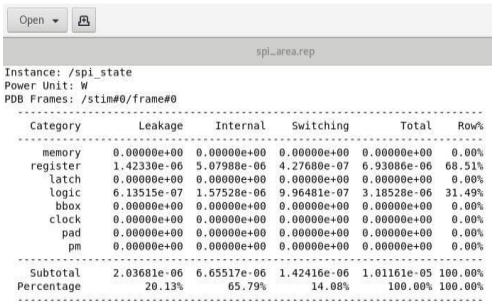

Fig 1.2: Component Power Rating Table

The power consumption of different components in an SPI area—likely a section of an integrated circuit—is displayed in the table. Five components are included in the table: memory, register, latch, and logic. As for Power Dissipation There are values for internal, switching, leakage, and total power dissipa- tion for every component. The units in which these numbers are most usually expressed are watts (W), yet the table does not do so. The table dissipates 1.01161 x 10^-5 Watts (W) of power in total, accord- ing to the image you submitted. Thus, 1.01161 x 10^- 5 becomes 0.0000101161 Watts when rewritten in conventional decimal notation. The circuit uses extremely little power, as indicated by this same value. Said another way, it indicates how energy-efficient the circuit is.

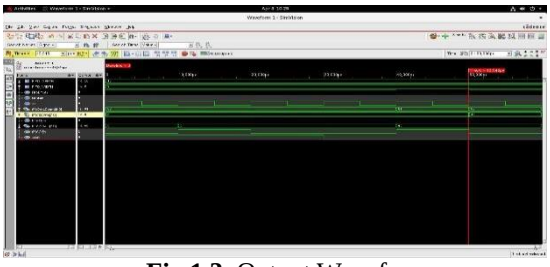

Fig 1.3: Output Waveform

A state machine is used by the spi_state module's State Machine to regulate the SPI signals.
States: State 0: Get ready for the next transmission, CS high, SCLK low.
State 1: decrease counter, set MOSI to the current data bit, CS low, and SCLK low.
State 2: shift out slightly, CS low, and SCLK high. On the spi_data line, data is sent serially, with the MSB arriving first. The counter keeps track of the total number of bits sent. The system clock, or clock, is what's used to synchronize. Bits of data are sent in 16 chunks. The SPI signals' timing is managed by the state machine.

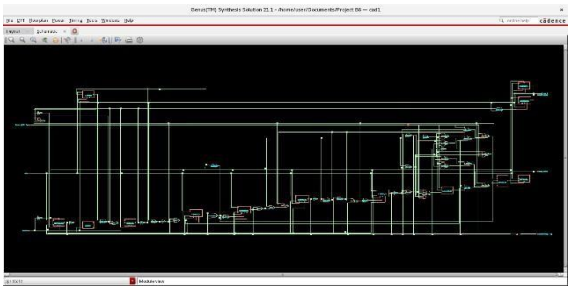

Fig 1.4: Schematic diagram

V. CONCLUSION

Using Verilog, we were able to develop the SPI (Serial Peripheral Interface) protocol for communication between a master device and several slave devices in this project. With a focus on the use of pipeline buffer and DMA technique, we also expanded the project to support communication in FPGA. Increased Flexibility of SPI protocol by allowing the master device to connect with several slave devices through a straightforward interface, offering a versatile communication option. We showed that various slaves might be chosen depending on the values of the bits using the slave selection process. Better Transmission Rate: SPI is renowned for its high-speed data transmission capacity, which is helpful for applications demanding quick data transfer. We were able to accomplish effective and quick communication between the master and slave devices by developing the protocol in Verilog. The SPI protocol, which depends on synchronous communication and clock signals, provided precise data delivery. This enhanced the system's overall dependability by removing potential data mistakes.

REFERENCES

[1]Design and Verification of Serial Peripheral Interface 1Ananthula Srinivas, 2M.Kiran Kumar , 3Jugal Kishore Bhandari, Int. National Jl of Engineering Development and Research, https://www.ijedr.org/papers/IJEDR1303026.pdf

[2]. Muhammad Bakir, Niju Rajan. VLSI implementation of SPI and I2C communication protocol. International Journal of Advance Research, Ideas and Innovations in technology 6 (2020).368-374

[3].Satish M Ghuse, Prof. Suresndra k, Waghmare FPGA implementation of I2C and SPI protocols using VHDL. IEEE Transactions on Very-Large-Scale Integration (VLSI) System.Int. National Jl.for Sci.and Advance Research in Technology.2(2016)16-20.

[4].Sukkwinder Kaur, Abhishek Godara, Amita Choudhary, 'Performance Analysis of Serial Peripheral Interface on FPGA',Int.National Jl. Of Engineering Research and technology 3(2014)

[5]. N. b. Mohd Noor and A. Saparon, "FPGA implementation of high speed serial peripheral interface for motion controller," 2012 IEEE Symposium on Industrial Electronics and Applications, *Bandung, Indonesia,* 2012, pp. 78-83, doi: 10.1109/ISIEA.2012.6496676.

[6] Jiayi Qiang et al ,FPGA Implementation of SPI Bus Communication Based on State Machine Method To cite this article: Jiayi Qiang et al 2020 J. Phys.: Conf. Ser. 1449 012027